



*National Information Exchange Model*

# ***Practical Implementer's Course***



United States  
Department of Justice

## **Implementation Strategies**



# Practical Implementer's Course



## In this section you will learn to...

- List main areas of interoperability concern in addition to Exchange Content
- Describe 2 main interface types for an IEPD
- Describe considerations to transform data presented in an IEPD into a form required for a native database, in both, web services and non web services technologies
- State considerations related to performance.



# Practical Implementer's Course



## Exchange Payload

- Is an important area in which to achieve interoperability
- NIEM helps by standardizing semantics & structure...
- Is only one of many areas that must be addressed
- ***By itself, does not guarantee interoperability***



# ***Practical Implementer's Course***



## **Areas of Interoperability Concern**

- Includes, but not limited to
  - Interface Definitions
  - Security
  - Messaging profiles



# Practical Implementer's Course



## Interface Considerations

- Close attention must be applied to the type of interface you choose to develop
  - Choices made can affect mapping and exchange construction effort/complexity
- 2 basic interface types
  - Tight
  - Loose



# Practical Implementer's Course



## Case 1: Tight Web Service Interface

- WSDL Includes:
  - Full operation description
    - Exchange document is explicitly visible in interface
    - Fully enumerated responses
- Service directly processes the exchange document



# Practical Implementer's Course



## Tight Coupling Example WSDL

- Consider the simple service for returning Incident Reports

```
<wsdl:portType name="IncidentRetrievalService">  
  <wsdl:operation name="retrieveIncidents">  
    <wsdl:input message="tns:retrieveIncidentsRequest"/>  
    <wsdl:output message="tns:retrieveIncidentResponse"/>  
    <wsdl:fault name="Error" message="tns:retrieveError"/>  
  </wsdl:operation>  
</wsdl:portType>
```

Return message brings  
back an Incident Report

Structure of response is  
defined in tns namespace



# Practical Implementer's Course



## Tight Coupling tns Namespace Schema

```
<xs:complexType name='IncidentResponseType'>
  <xs:complexContent>
    <xs:extension base='svc:ServiceResponseConfirmation' />
    <xs:choice>
      <xs:element ref='noRecord' minOccurs='1' maxOccurs='1' />
      <xs:element
        xmlns:iep='http://www.niem.gov/jxdm/doc/incident/1.0/document'
        ref='iep:IncidentReport' minOccurs='1' maxOccurs='1' />
    </xs:choice>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:element name="retrievedIncidentResponse" type="rec:IncidentResponseType"/>
```

Embeds IncidentReport IEPD directly into response message

Response Structure referred to by WSDL





# ***Practical Implementer's Course***



## **Pros**

- Very explicit definition
- Easy to support discovery
- Web service validates exchange document since it is part of the WSDL



# Practical Implementer's Course



## Cons

- Must use concrete typed extensions for Exchange
  - WSDL does not support type substitution
  - Larger work effort producing cascaded extensions
- WSDLs tend to be complex
- Exchange document changes require WSDL changes
  - Can affect service stakeholders



# Practical Implementer's Course



## Approach 2: Loose Web Service Interface

- WSDL Includes:
  - Messaging operation description only
    - Exchange document is not explicitly visible in interface
      - Usually contains an <xs:any> element for the exchange document
    - Fully enumerated responses
- Implies secondary processing behind the web service to process/handle the exchange document itself



# Practical Implementer's Course



## Loose Coupling Example WSDL

- Consider the simple service for returning Incident Reports

```
<wsdl:portType name="RecordRetrievalService">  
  <wsdl:operation name="retrieveRecords">  
    <wsdl:input message="tns:retrieveRecordsRequest"/>  
    <wsdl:output message="tns:retrieveRecordResponse"/>  
    <wsdl:fault name="Error" message="tns:retrieveError"/>  
  </wsdl:operation>  
</wsdl:portType>
```

Return message brings  
back an Incident Report

Structure of response is  
defined in tns namespace



# Practical Implementer's Course



## Loose Coupling tns Namespace Schema

```
<xs:complexType name='RecordResponseType'>
  <xs:complexContent>
    <xs:extension base='svc:ServiceResponseConfirmationType'>
      <xs:choice>
        <xs:element ref='noRecord' minOccurs='1' maxOccurs='1' />
        <xs:any/>
      </xs:choice>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="retrievedRecordResponse" type="rec:RecordResponseType"/>
```

Accepts any content as valid.  
Requires secondary processing to  
determine validity of payload

Response Structure  
referred to by WSDL



# Practical Implementer's Course



## Pros

- Simple WSDL
- Straightforward interface
- Flexible
- Allows upgrades without redeploying web service
- Simpler extensions
  - Can use type substitution because the Exchange Document is abstracted out of the WSDL
- Easily supports multiple versions/types of Exchange Documents



# Practical Implementer's Course



## Cons

- Implies a requirement for versioning and supported exchange discovery in Interface (WSDL)
  - Work to implement
  - More complicated discovery
- Requires extra processing after the Web Service to process the Exchange Document



# Practical Implementer's Course



## Interface Conclusions

- Neither way is better
- Choice is dependent of overall project requirements
  - Process integration tends to define rigid interfaces
  - Search oriented integrations tend to define loose interfaces
- Each approach has development implications and must be applied at the project level





# Practical Implementer's Course



## IEPD to Database Transformation

- At some point, IEP based data is coming from or going to a database
- Transforming into a database schema rarely reflects the IEP
  - Beware! Some data needed for database may be implicit in structure as opposed to explicit data value
    - Indicators and dates are a good example for this
- Spreadsheets are good to use here
  - Use IEP Mapping spreadsheet as baseline or template for your own database specific mapping document



# Practical Implementer's Course

## Performance Considerations

- Ultimately a requirement of sharing project and its intended use
- Validation is easily the largest real-time consumer
  - Can be mitigated by mandating certification levels
  - Validation relegated to testing phase in development environments



# Practical Implementer's Course



## Summary

- Interoperability
  - Exchange documents are only one piece of the puzzle
- Interfaces
  - Rigid & Loose
- Database to NIEM Mapping approach
- Performance Considerations



# Practical Implementer's Course



This work is licensed under the Creative Commons Attribution-ShareAlike 2.5 License.

To view a copy of this license

- a) visit <http://creativecommons.org/licenses/by-sa/2.5/>; or,
- b) send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA."